arXiv:1908.07873v1 [cs.LG] 21 Aug 2019

Federated Learning: Challenges, Methods, and Future Directions

Tian Li
Carnegie Mellon University
tianli@cmu.edu

Anit Kumar Sahu
Bosch Center for Artificial Intelligence
anit.sahu@gmail.com

Ameet Talwalkar
Carnegie Mellon University & Determined AI
talwalkar@cmu.edu

Virginia Smith
Carnegie Mellon University
smithv@cmu.edu

Abstract

Federated learning involves training statistical models over remote devices or siloed data centers, such as mobile phones or hospitals, while keeping data localized. Training in heterogeneous and potentially massive networks introduces novel challenges that require a fundamental departure from standard approaches for large-scale machine learning, distributed optimization, and privacy-preserving data analysis. In this article, we discuss the unique characteristics and challenges of federated learning, provide a broad overview of current approaches, and outline several directions of future work that are relevant to a wide range of research communities.

1 Introduction

Mobile phones, wearable devices, and autonomous vehicles are just a few of the modern distributed networks generating a wealth of data each day. Due to the growing computational power of these devices—coupled with concerns over transmitting private information—it is increasingly attractive to store data *locally* and push network computation to the edge.

The concept of edge computing is not a new one. Indeed, computing simple queries across distributed, low-powered devices is a decades-long area of research that has been explored under the purview of query processing in sensor networks, computing at the edge, and fog computing [12, 29, 40, 49, 74]. Recent works have also considered training machine learning models centrally but serving and storing them locally; for example, this is a common approach in mobile user modeling and personalization [60, 90].

However, as the storage and computational capabilities of the devices within distributed networks grow, it is possible to leverage enhanced local resources on each device. This has led to a growing interest in *federated learning* [75], which explores *training* statistical models directly on remote devices[1]. As we discuss in this article, learning in such a setting differs significantly from traditional distributed environments—requiring

[1]We use the term 'device' throughout the article to describe entities in the network, such as nodes, clients, sensors, or organizations.

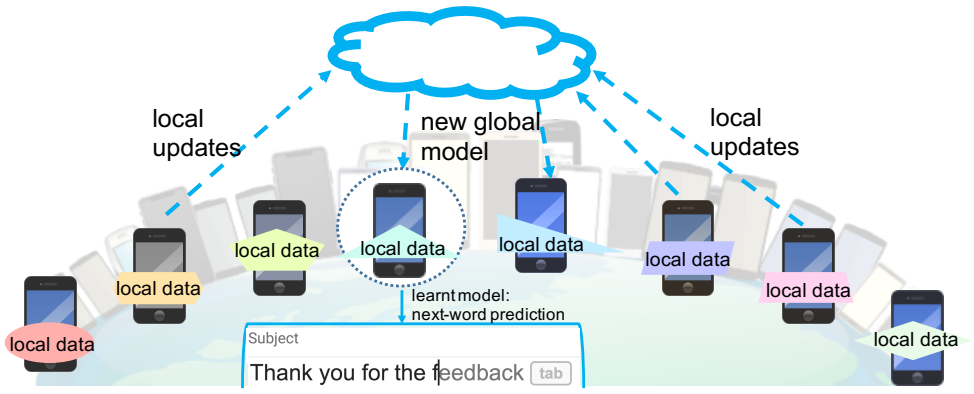


Figure 1: An example application of federated learning for the task of next-word prediction on mobile phones. To preserve the privacy of the text data and to reduce strain on the network, we seek to train a predictor in a distributed fashion, rather than sending the raw data to a central server. In this setup, remote devices communicate with a central server periodically to learn a global model. At each communication round, a subset of selected phones performs local training on their non-identically-distributed user data, and sends these local updates to the server. After incorporating the updates, the server then sends back the new global model to another subset of devices. This iterative training process continues across the network until convergence is reached or some stopping criterion is met.

fundamental advances in areas such as privacy, large-scale machine learning, and distributed optimization, and raising new questions at the intersection of diverse fields, such as machine learning and systems [91].

Federated learning methods have been deployed by major service providers [11, 124], and play a critical role in supporting privacy-sensitive applications where the training data are distributed at the edge [e.g., 5, 46, 51, 89, 105, 127, 139]. Examples of potential applications include: learning sentiment, semantic location, or activities of mobile phone users; adapting to pedestrian behavior in autonomous vehicles; and predicting health events like heart attack risk from wearable devices [6, 52, 84]. We discuss several canonical applications of federated learning below:

- *Smart phones.* By jointly learning user behavior across a large pool of mobile phones, statistical models can power applications such as next-word prediction, face detection, and voice recognition [46, 89]. However, users may not be willing to share their data in order to protect their personal privacy or to save the limited bandwidth/battery power of their phone. Federated learning has the potential to enable predictive features on smart phones without diminishing the user experience or leaking private information. Figure 1 depicts one such application in which we aim to learn a next-word predictor in a large-scale mobile phone network based on users' historical text data [46].

- *Organizations.* Organizations or institutions can also be viewed as 'devices' in the context of federated learning. For example, hospitals are organizations that contain a multitude of patient data for predictive healthcare. However, hospitals operate under strict privacy practices, and may face legal, administrative, or ethical constraints that require data to remain local. Federated learning is a promising solution for these applications [52], as it can reduce strain on the network and enable private learning between various devices/organizations.

- *Internet of things.* Modern IoT networks, such as wearable devices, autonomous vehicles, or smart homes, may contain numerous sensors that allow them to collect, react, and adapt to incoming data in real-time. For example, a fleet of autonomous vehicles may require an up-to-date model of traffic,

construction, or pedestrian behavior to safely operate. However, building aggregate models in these scenarios may be difficult due to the private nature of the data and the limited connectivity of each device. Federated learning methods can help to train models that efficiently adapt to changes in these systems while maintaining user privacy [84, 98].

1.1 Problem Formulation

The canonical federated learning problem involves learning a *single, global* statistical model from data stored on tens to potentially millions of remote devices. We aim to learn this model under the constraint that device-generated data is stored and processed locally, with only intermediate updates being communicated periodically with a central server. In particular, the goal is typically to minimize the following objective function:

$$\min_w F(w), \quad \text{where} \quad F(w) := \sum_{k=1}^{m} p_k F_k(w). \tag{1}$$

Here, m is the total number of devices, $p_k \geq 0$ and $\sum_k p_k = 1$, and F_k is the local objective function for the kth device. The local objective function is often defined as the empirical risk over local data, i.e., $F_k(w) = \frac{1}{n_k} \sum_{j_k=1}^{n_k} f_{j_k}(w; x_{j_k}, y_{j_k})$, where n_k is the number of samples available locally. The user-defined term p_k specifies the relative impact of each device, with two natural settings being $p_k = \frac{1}{n}$ or $p_k = \frac{n_k}{n}$, where $n = \sum_k n_k$ is the total number of samples. We will reference problem (1) throughout the article, but, as discussed below, we note that other objectives or modeling approaches may be appropriate depending on the application of interest.

1.2 Core Challenges

We next describe four of the core challenges associated with solving the distributed optimization problem posed in (1). These challenges make the federated setting distinct from other classical problems, such as distributed learning in data center settings or traditional private data analyses.

Challenge 1: Expensive Communication. Communication is a critical bottleneck in federated networks, which, coupled with privacy concerns over sending raw data, necessitates that data generated on each device remain local. Indeed, federated networks are potentially comprised of a massive number of devices, e.g., millions of smart phones, and communication in the network can be slower than local computation by many orders of magnitude [50, 115]. In order to fit a model to data generated by the devices in the federated network, it is therefore necessary to develop communication-efficient methods that iteratively send small messages or *model updates* as part of the training process, as opposed to sending the entire dataset over the network. To further reduce communication in such a setting, two key aspects to consider are: (i) reducing the total number of communication rounds, or (ii) reducing the size of transmitted messages at each round.

Challenge 2: Systems Heterogeneity. The storage, computational, and communication capabilities of each device in federated networks may differ due to variability in hardware (CPU, memory), network connectivity (3G, 4G, 5G, wifi), and power (battery level). Additionally, the network size and systems-related constraints on each device typically result in only a small fraction of the devices being active at once, e.g., hundreds of active devices in a million-device network [11]. Each device may also be unreliable, and it is not uncommon for an active device to drop out at a given iteration due to connectivity or energy constraints.

These system-level characteristics dramatically exacerbate challenges such as straggler mitigation and fault tolerance. Federated learning methods that are developed and analyzed must therefore: (i) anticipate a low amount of participation, (ii) tolerate heterogeneous hardware, and (iii) be robust to dropped devices in the network.

Challenge 3: Statistical Heterogeneity. Devices frequently generate and collect data in a non-identically distributed manner across the network, e.g., mobile phone users have varied use of language in the context of a next word prediction task. Moreover, the number of data points across devices may vary significantly, and there may be an underlying structure present that captures the relationship amongst devices and their associated distributions. This data generation paradigm violates frequently-used independent and identically distributed (I.I.D.) assumptions in distributed optimization, increases the likelihood of stragglers, and may add complexity in terms of modeling, analysis, and evaluation. Indeed, although the canonical federated learning problem of (1) aims to learn a single global model, there exist other alternatives such as simultaneously learning distinct local models via multi-task learning frameworks [cf. 106]. There is also a close connection in this regard between leading approaches for federated learning and meta-learning [64]. Both the multi-task and meta-learning perspectives enable *personalized* or *device-specific* modeling, which is often a more natural approach to handle the statistical heterogeneity of the data.

Challenge 4: Privacy Concerns. Finally, privacy is often a major concern in federated learning applications. Federated learning makes a step towards protecting data generated on each device by sharing model updates, e.g., gradient information, instead of the raw data [17, 31, 33]. However, communicating model updates throughout the training process can nonetheless reveal sensitive information, either to a third-party, or to the central server [76]. While recent methods aim to enhance the privacy of federated learning using tools such as secure multiparty computation or differential privacy, these approaches often provide privacy at the cost of reduced model performance or system efficiency. Understanding and balancing these trade-offs, both theoretically and empirically, is a considerable challenge in realizing private federated learning systems.

The remainder of this article is organized as follows. In Section 2, we introduce previous and current works that aim to address the four discussed challenges of federated learning. In Section 3, we outline several promising directions of future research.

2 Survey of Related and Current Work

The challenges in federated learning at first glance resemble classical problems in areas such as privacy, large-scale machine learning, and distributed optimization. For instance, numerous methods have been proposed to tackle expensive communication in the machine learning, optimization, and signal processing communities. However, these methods are typically unable to fully handle the scale of federated networks, much less the challenges of systems and statistical heterogeneity. Similarly, while privacy is an important aspect for many machine learning applications, privacy-preserving methods for federated learning can be challenging to rigorously assert due to the statistical variation in the data, and may be even more difficult to implement due to systems constraints on each device and across the potentially massive network. In this section, we explore in more detail the challenges presented in Section 1, including a discussion of classical results as well as more recent work focused specifically on federated learning.

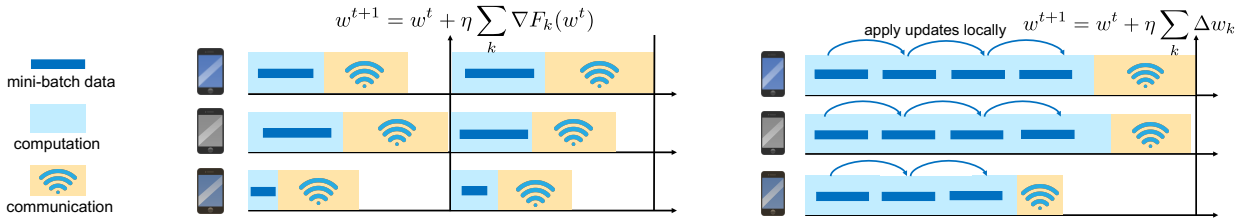


Figure 2: *Left: Distributed (mini-batch) SGD.* Each device, k, locally computes gradients from a mini-batch of data points to approximate $\nabla F_k(w)$, and the aggregated mini-batch updates are applied on the server. *Right: Local updating schemes.* Each device immediately applies local updates, e.g., gradients, after they are computed and a server performs a global aggregation after a variable number of local updates. Local-updating schemes can reduce communication by performing additional work locally.

2.1 Communication-efficiency

Communication is a key bottleneck to consider when developing methods for federated networks. While it is beyond the scope of this article to provide a self-contained review of communication-efficient distributed learning methods, we point out several general directions, which we group into (1) local updating methods, (2) compression schemes, and (3) decentralized training.

2.1.1 Local Updating

Mini-batch optimization methods, which involve extending classical stochastic methods to process multiple data points at a time, have emerged as a popular paradigm for distributed machine learning in data center environments [28, 88, 96, 102, 103]. In practice, however, they have been shown to have limited flexibility to adapt to communication-computation trade-offs that would maximally leverage distributed data processing [107, 108]. In response, several recent methods have been proposed to improve communication-efficiency in distributed settings by allowing for a variable number of *local updates* to be applied on each machine in parallel at each communication round, making the amount of computation versus communication substantially more flexible. For convex objectives, distributed local-updating *primal-dual* methods have emerged as a popular way to tackle such a problem [54, 62, 72, 107, 128]. These approaches leverage duality structure to effectively decompose the global objective into subproblems that can be solved in parallel at each communication round. Several distributed local-updating *primal* methods have also been proposed, which have the added benefit of being applicable to non-convex objectives [93, 136]. These methods drastically improve performance in practice, and have been shown to achieve orders-of-magnitude speedups over traditional mini-batch methods or distributed approaches like ADMM [14] in real-world data center environments. We provide an intuitive illustration of local updating methods in Figure 2.

In federated settings, optimization methods that allow for flexible local updating and low client participation have become the de facto solvers [65, 75, 106]. The most commonly used method for federated learning is Federated Averaging (`FedAvg`) [75], a method based on averaging local stochastic gradient descent (SGD) updates for the primal problem. `FedAvg` has been shown to work well empirically, particularly for non-convex problems, but comes without convergence guarantees and can diverge in practical settings when data are heterogeneous [65]. We discuss methods to handle such statistical heterogeneity in more detail in Section 2.3.2.

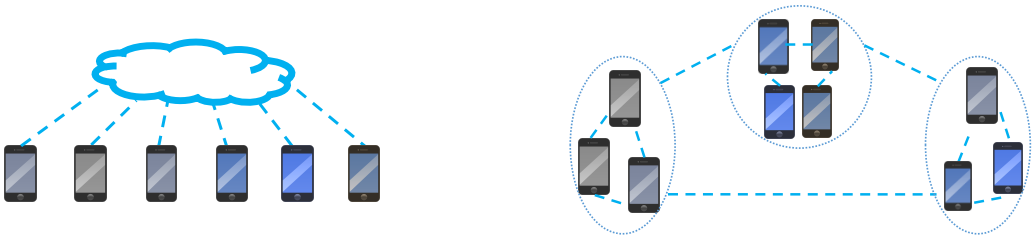


Figure 3: Centralized vs. decentralized topologies. In the typical federated learning setting and as a focus of this article, we assume a star network (left) where a server connects with all remote devices. Decentralized topologies (right) are a potential alternative when communication to the server becomes a bottleneck.

2.1.2 Compression Schemes

While local updating methods can reduce the total *number of communication rounds*, model compression schemes such as sparsification, subsampling, and quantization can significantly reduce the *size of messages* communicated at each round. These methods have been extensively studied, both empirically and theoretically, in previous literature for distributed training in data center environments; we defer the readers to [119, 135] for a more complete review. In federated environments, the low participation of devices, non-identically distributed local data, and local updating schemes pose novel challenges to these model compression approaches. For instance, the commonly-used error compensation techniques in classical distributed learning [101] cannot be directly extended to federated settings as the errors accumulated locally may be stale if the devices are not frequently sampled. Nevertheless, several works have provided practical strategies in federated settings, such as forcing the updating models to be sparse and low-rank; performing quantization with structured random rotations [59]; using lossy compression and dropout to reduce server-to-device communication [15]; and applying Golomb lossless encoding [99]. From a theoretical perspective, while prior work has explored convergence guarantees with low-precision training in the presence of non-identically distributed data [e.g., 111], the assumptions made do not take into consideration common characteristics of the federated setting, such as low device participation or locally-updating optimization methods.

2.1.3 Decentralized Training

In federated learning, a star network (where a central server is connected to a network of devices, as in the left panel of Figure 3) is the predominant communication topology; we therefore focus on the star-network setting in this article. However, we briefly discuss decentralized topologies (where devices only communicate with their neighbors, e.g., the right panel of Figure 3) as a potential alternative. In data center environments, decentralized training has been demonstrated to be faster than centralized training when operating on networks with low bandwidth or high latency; we defer readers to [47, 67] for a more comprehensive review. Similarly, in federated learning, decentralized algorithms can in theory reduce the high communication cost on the central server. Some recent works [47, 61] have investigated decentralized training over heterogeneous data with local updating schemes. However, they are either restricted to linear models [47] or assume full device participation [61]. Finally, hierarchical communication patterns have also been proposed [68, 70] to further ease the burden on the central server, by first leveraging *edge servers* to aggregate the updates from edge devices and then relying on a *cloud server* to aggregate updates from edge servers. While this is a promising approach to reduce communication, it is not applicable to all networks, as this type of physical hierarchy may not exist or be known a priori.

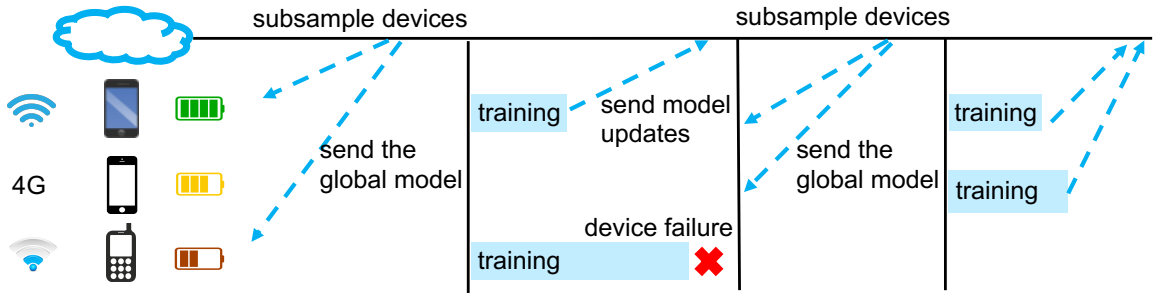


Figure 4: Systems heterogeneity in federated learning. Devices may vary in terms of network connection, power, and hardware. Moreover, some of the devices may drop at any time during training. Therefore, federated training methods must tolerate heterogeneous systems environments and low participation of devices, i.e., they must allow for only a small subset of devices to be active at each round.

2.2 Systems Heterogeneity

In federated settings, there is significant variability in the *systems* characteristics across the network, as devices may differ in terms of hardware, network connectivity, and battery power. As depicted in Figure 4, these systems characteristics make issues such as stragglers significantly more prevalent than in typical data center environments. We roughly group several key directions to handle systems heterogeneity into: (i) asynchronous communication, (ii) active device sampling, and (ii) fault tolerance. As mentioned in Section 2.1.3, we assume a star topology in our following discussions.

2.2.1 Asynchronous Communication

In traditional data center settings, synchronous and asynchronous schemes are both commonly used to parallelize iterative optimization algorithms, with each approach having pros and cons. Synchronous schemes are simple and guarantee a serial-equivalent computational model, but they are also more susceptible to stragglers in the face of device variability. Asynchronous schemes are an attractive approach to mitigate stragglers in heterogeneous environments, particularly in shared-memory systems [27, 30, 48, 92, 141]. However, they typically rely on bounded-delay assumptions to control the degree of staleness, which for device k depends on the number of other devices that have updated since device k pulled from the central server. While asynchronous parameter servers have been successful in distributed data centers [e.g., 27, 48, 141], classical bounded-delay assumptions can be unrealistic in federated settings, where the delay may be on the order of hours to days, or completely unbounded.

2.2.2 Active Sampling

In federated networks, typically only a small subset of devices participate at each round of training. However, the vast majority of federated methods, e.g. those described in [11, 47, 65, 75, 106], are *passive* in that they do not aim to influence which devices participate. An alternative approach involves *actively* selecting participating devices at each round. For example, Nishio and Yonetani [83] explore novel device sampling policies based on systems resources, with the aim being for the server to aggregate as many device updates as possible within a pre-defined time window. Similarly, Kang et al. [57] take into account

systems overheads incurred on each device when designing incentive mechanisms to encourage devices with higher-quality data to participate in the learning process. However, these methods assume a static model of the systems characteristics of the network; it remains open how to extend these approaches to handle *real-time*, device-specific fluctuations in computation and communication delays. Moreover, while these methods primarily focus on systems variability to perform active sampling, we note that it is also worth considering actively sampling a set of small but sufficiently representative devices based on the underlying *statistical* structure.

2.2.3 Fault Tolerance

Fault tolerance has been extensively studied in the systems community and is a fundamental consideration of classical distributed systems [19, 71, 110]. Recent works have also investigated fault tolerance specifically for machine learning workloads in data center environments [e.g., 87, 112]. When learning over remote devices, however, fault tolerance becomes more critical as it is common for some participating devices to drop out at some point before the completion of the given training iteration [11]. One practical strategy is to simply ignore such device failure [11], which may introduce bias into the device sampling scheme if the failed devices have specific data characteristics. For instance, devices from remote areas may be more likely to drop due to poor network connections and thus the trained federated model will be biased towards devices with favorable network conditions. Theoretically, while several recent works have investigated convergence guarantees of variants of federated learning methods [56, 123, 131, 132], few analyses allow for low participation [e.g., 65, 106], or study directly the effect of dropped devices.

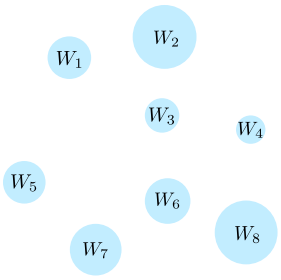
Coded computation is another option to tolerate device failures by introducing algorithmic redundancy. Recent works have explored using codes to speed up distributed machine learning training [e.g., 20, 21, 63, 94, 109]. For instance, in the presence of stragglers, gradient coding and its variants [20, 21, 109] carefully replicate data blocks (as well as the gradient computation on those data blocks) across computing nodes to obtain either exact or inexact recovery of the true gradients. While this is a seemingly promising approach for the federated setting, these methods face fundamental challenges in federated networks as sharing data/replication across devices is often infeasible due to privacy constraints and the scale of the network.

2.3 Statistical Heterogeneity

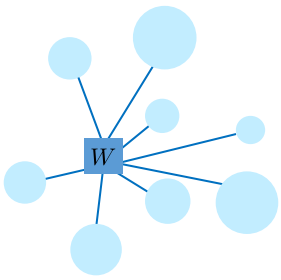
Challenges arise when training federated models from data that is not identically distributed across devices, both in terms of modeling the data (as depicted in Figure 5), and in terms of analyzing the convergence behavior of associated training procedures. We discuss related work in these directions below.

2.3.1 Modeling Heterogeneous Data

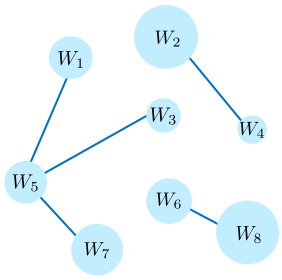
There exists a large body of literature in machine learning that has modeled statistical heterogeneity via methods such as meta-learning [114] and multi-task learning [18, 37]; these ideas have been recently extended to the federated setting [24, 26, 35, 58, 106, 138]. For instance, `MOCHA` [106], an optimization framework designed for the federated setting, can allow for personalization by learning *separate* but related models for each device while leveraging a shared representation via multi-task learning. This method has provable theoretical convergence guarantees for the considered objectives, but is limited in its ability to



(a) Learn personalized models for each device; do not learn from peers.



(b) Learn a global model; learn from peers.



(c) Learn personalized models for each device; learn from peers.

Figure 5: Different modeling approaches in federated networks. Depending on properties of the data, network, and application of interest, one may choose to (a) learn separate models for each device, (b) fit a single global model to all devices, or (c) learn related but distinct models in the network.

scale to massive networks and is restricted to convex objectives. Another approach [26] models the star topology as a Bayesian network and performs variational inference during learning. Although this method can handle non-convex models, it is expensive to generalize to large federated networks. Khodak et al. [58] provably meta-learn a within-task learning rate using multi-task information (where each task corresponds to a device) and have demonstrated improved empirical performance over vanilla `FedAvg`. Eichner et al. [35] investigate a pluralistic solution (adaptively choosing between a global model and device-specific models) to address the cyclic patterns in data samples during federated training. Zhao et al. [138] explore transfer learning for personalization by running `FedAvg` after training a global model centrally on some shared proxy data. Despite these recent advances, key challenges still remain in making methods for heterogeneous modeling that are robust, scalable, and automated in federated settings.

When modeling federated data, it may also be important to consider issues beyond accuracy, such as *fairness*. In particular, naively solving an aggregate loss function such as in (1) may implicitly advantage or disadvantage some of the devices, as the learned model may become biased towards devices with larger amounts of data, or (if weighting devices equally), to commonly occurring groups of devices. Recent works have proposed modified modeling approaches that aim to reduce the variance of the model performance across devices. Some heuristics simply perform a varied number of local updates based on local loss [52]. Other more principled approaches include Agnostic Federated Learning [80], which optimizes the centralized model for any target distribution formed by a mixture of the client distributions via a minimax optimization scheme. Another more general approach is taken by Li et al. [66], which proposes an objective called q-FFL in which devices with higher loss are given higher relative weight to encourage less variance in the final accuracy distribution. Beyond issues of fairness, we note that aspects such as accountability and interpretability in federated learning are additionally worth exploring, but may be challenging due to the scale and heterogeneity of the network.

2.3.2 Convergence Guarantees for Non-IID Data

Statistical heterogeneity also presents novel challenges in terms of analyzing the convergence behavior in federated settings—even when learning a single global model. Indeed, when data is not identically distributed across devices in the network, methods such as `FedAvg` have been shown to diverge in practice [65, 75]. Parallel SGD and related variants, which make local updates similar to `FedAvg`, have been

analyzed in the I.I.D. setting [68, 93, 104, 108, 120, 121, 122, 125, 136, 140]. However, the results rely on the premise that each local solver is a copy of the same stochastic process (due to the I.I.D. assumption), which is not the case in typical federated settings. To understand the performance of `FedAvg` in statistically heterogeneous settings, `FedProx` [65] has recently been proposed. `FedProx` makes a small modification to the `FedAvg` method to help ensure convergence, both theoretically and in practice. `FedProx` can also be interpreted as a generalized, reparameterized version of `FedAvg` that has practical ramifications in the context of accounting for systems heterogeneity across devices. Several other works [56, 123, 131, 132] have also explored convergence guarantees in the presence of heterogeneous data with different assumptions, e.g., convexity [123] or uniformly bounded gradients [131]. There are also heuristic approaches that aim to tackle statistical heterogeneity, either by sharing local device data or some server-side proxy data [52, 55, 138]. However, these methods may be unrealistic: in addition to imposing burdens on network bandwidth, sending local data to the server [55] violates the key privacy assumption of federated learning, and sending globally-shared proxy data to all devices [52, 138] requires effort to carefully generate or collect such auxiliary data.

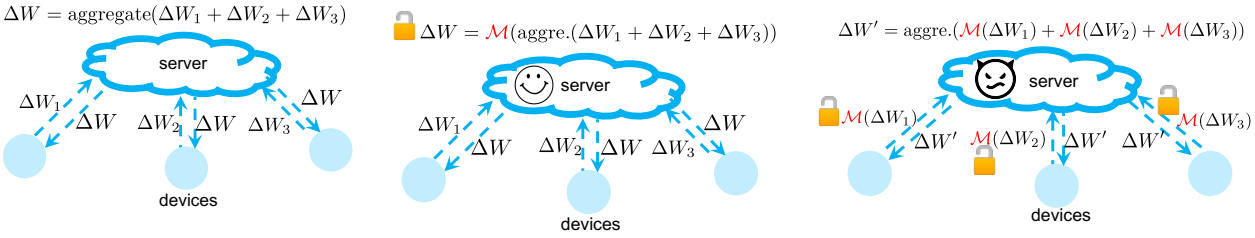
2.4 Privacy

Privacy concerns often motivate the need to keep raw data on each device local in federated settings. However, sharing other information such as model updates as part of the training process can also leak sensitive user information [8, 17, 39, 78]. For instance, Carlini et al. [17] demonstrate that one can extract sensitive text patterns, e.g., a specific credit card number, from a recurrent neural network trained on users' language data. Given increasing interest in privacy-preserving learning approaches, in Section 2.4.1, we first briefly revisit prior work on enhancing privacy in the general (distributed) machine learning setting. We then review recent privacy-preserving methods specifically designed for federated settings in Section 2.4.2.

2.4.1 Privacy in Machine Learning

Privacy-preserving learning has been extensively studied by the machine learning [e.g., 76], systems [e.g., 4, 11], and theory [e.g., 38, 69] communities. Three main strategies, each of which we will briefly review, include differential privacy to communicate noisy data sketches, homomorphic encryption to operate on encrypted data, and secure function evaluation or multiparty computation.

Among these various privacy approaches, *differential privacy* [32, 33, 34] is most widely used due to its strong information theoretic guarantees, algorithmic simplicity, and relatively small systems overhead. Simply put, a randomized mechanism is differentially private if the change of one input element will not result in too much difference in the output distribution; this means that one cannot draw any conclusions about whether or not a specific sample is used in the learning process. Such sample-level privacy can be achieved in many learning tasks [2, 7, 22, 53, 85, 86]. For gradient-based learning methods, a popular approach is to apply differential privacy by randomly perturbing the intermediate output at each iteration [e.g., 2, 7, 126]. Before applying the perturbation, e.g., via Gaussian noise [2], Laplacian noise [77], or Binomial noise [3], it is common to clip the gradients in order to bound the influence of each example on the overall update. There exists an inherent trade-off between differential privacy and model accuracy, as adding more noise results in greater privacy, but may compromise accuracy significantly. Despite the fact that differential privacy is the de facto metric for privacy in machine learning, there are many other privacy definitions, such as k-anonymity [36], δ-presence [81] and distance correlation [117], that may be applicable to different learning problems [118].



(a) Federated learning without additional privacy protection mechanisms.

(b) Global privacy, where a trusted server is assumed.

(c) Local privacy, where the central server might be malicious.

Figure 6: An illustration of different privacy-enhancing mechanisms in one round of federated learning. $\mathcal{M}$ denotes a randomized mechanism used to privatize the data. With global privacy (b), the model updates are private to all third parties other than a single trusted party (the central server). With local privacy (c), the individual model updates are also private to the server.

Beyond differential privacy, homomorphic encryption can be used to secure the learning process by computing on encrypted data, although it has currently been applied in limited settings, e.g., training linear models [82] or involving only a few entities [133]. When the sensitive datasets are distributed across different data owners, another natural option is to perform privacy-preserving learning via secure function evaluation (SFE) or secure multiparty computation (SMC). The resulting protocols can enable multiple parties to collaboratively compute an agreed-upon function without leaking input information from any party except for what can be inferred from the output [e.g., 23, 43, 95]. Thus, while SMC cannot guarantee protection from information leakage, it can be combined with differential privacy to achieve stronger privacy guarantees. However, approaches along these lines may not be applicable to large-scale machine learning scenarios as they incur substantial additional communication and computation costs. Moreover, SMC protocols need to be carefully designed and implemented for each operation in the targeted learning algorithm [25, 79]. We defer interested readers to [13, 97] for a more comprehensive review of the approaches based on homomorphic encryption and SMC.

2.4.2 Privacy in Federated Learning

The federated setting poses novel challenges to existing privacy-preserving algorithms. Beyond providing rigorous privacy guarantees, it is necessary to develop methods that are computationally cheap, communication-efficient, and tolerant to dropped devices—all without overly compromising accuracy. Although there are a variety of privacy definitions in federated learning [8, 17, 41, 64, 76, 113], typically they can be classified into two categories: *global privacy* and *local privacy*. As demonstrated in Figure 6, global privacy requires that the model updates generated at each round are private to all untrusted third parties other than the central server, while local privacy further requires that the updates are also private to the server.

Current works that aim to improve the privacy of federated learning typically build upon previous classical cryptographic protocols such as SMC [10, 42] and differential privacy [3, 8, 41, 76]. Bonawitz et al. [10] introduce an SMC protocol to protect individual model updates. The central server is not able to see any local updates, but can still observe the exact aggregated results at each round. SMC is a lossless method, and can retain the original accuracy with a very high privacy guarantee. However, the resulting method incurs significant extra communication cost. Other works [41, 76] apply differential privacy to federated learning

and offer global differential privacy. These approaches have a number of hyperparameters that affect communication and accuracy that must be carefully chosen, though follow up work [113] proposes adaptive gradient clipping strategies to help alleviate this issue. In the case where stronger privacy guarantees are required, Bhowmick et al. [8] introduce a relaxed version of local privacy by limiting the power of potential adversaries. It affords stronger privacy guarantees than global privacy, and has better model performance than strict local privacy. Li et al. [64] propose locally differentially-private algorithms in the context of meta-learning, which can be applied to federated learning with personalization, while also providing provable learning guarantees in convex settings. In addition, differential privacy can be combined with model compression techniques to reduce communication and obtain privacy benefits simultaneously [3].

3 Future Directions

Federated learning is an active and ongoing area of research. Although recent work has begun to address the challenges discussed in Section 2, there are a number of critical open directions yet to be explored. In this section, we briefly outline a few promising research directions surrounding the previously discussed challenges (expensive communication, systems heterogeneity, statistical heterogeneity, and privacy concerns), and introduce additional challenges regarding issues such as productionizing and benchmarking in federated settings.

- **Extreme communication schemes.** It remains to be seen how much communication is necessary in federated learning. Indeed, it is well-known that optimization methods for machine learning can tolerate a lack of precision; this error can in fact help with generalization [129]. While one-shot or divide-and-conquer communication schemes have been explored in traditional data center settings [73, 137], the behavior of these methods is not well-understood in massive or statistical heterogeneous networks. Similarly, one-shot/few-shot heuristics [44, 45, 134] have recently been proposed for the federated setting, but have yet to be theoretically analyzed or evaluated at scale.
- **Communication reduction and the Pareto frontier.** We discussed several ways to reduce communication in federated training, such as local updating and model compression. In order to create a realistic system for federated learning, it is important to understand how these techniques *compose* with one another, and to *systematically* analyze the trade-off between accuracy and communication for each approach. In particular, the most useful techniques will demonstrate improvements at the Pareto frontier—achieving an accuracy greater than any other approach under the same communication budget, and ideally, across a wide range of communication/accuracy profiles. Similar comprehensive analyses have been performed for efficient neural network inference [e.g., 9], and are necessary in order to compare communication-reduction techniques for federated learning in a meaningful way.
- **Novel models of asynchrony.** As discussed in Section 2.2.1, two communication schemes most commonly studied in distributed optimization are bulk synchronous approaches and asynchronous approaches (where it is assumed that the delay is bounded). These schemes are more realistic in data center settings—where worker nodes are typically *dedicated* to the workload, i.e., they are ready to 'pull' their next job from the central node immediately after they 'push' the results of their previous job. In contrast, in federated networks, each device is often *undedicated* to the task at hand and most devices are not active on any given iteration. Therefore, it is worth studying the effects of this more realistic *device-centric* communication scheme—in which each device can decide when to 'wake up' and interact with the central server in an event-triggered manner.

- **Heterogeneity diagnostics.** Recent works have aimed to quantify statistical heterogeneity through metrics such as local dissimilarity (as defined in the context of federated learning in [65] and used for other purposes in works such as [100, 116, 130]) and earth mover's distance [138]. However, these metrics cannot be easily calculated over the federated network before training occurs. The importance of these metrics motivates the following open questions: (i) Do simple diagnostics exist to quickly determine the level of heterogeneity in federated networks *a priori*? (ii) Can analogous diagnostics be developed to quantify the amount of *systems-related* heterogeneity? (iii) Can current or new definitions of heterogeneity be exploited to further improve the convergence of federated optimization methods?

- **Granular privacy constraints.** The definitions of privacy outlined in Section 2.4.2 cover privacy at a local or global level with respect to all devices in the network. However, in practice, it may be necessary to define privacy on a more granular level, as privacy constraints may differ across devices or even across data points on a single device. For instance, Li et al. [64] recently proposed sample-specific (as opposed to user-specific) privacy guarantees, thus providing a weaker form of privacy in exchange for more accurate models. Developing methods to handle mixed (device-specific or sample-specific) privacy restrictions is an interesting and ongoing direction of future work.

- **Beyond supervised learning.** It is important to note that the methods discussed thus far have been developed with the task of *supervised learning* in mind, i.e., they assume that labels exist for all of the data in the federated network. In practice, much of the data generated in realistic federated networks may be unlabeled or weakly labeled. Furthermore, the problem at hand may not be to fit a model to data as presented in (1), but instead to perform some exploratory data analysis, determine aggregate statistics, or run a more complex task such as reinforcement learning. Tackling problems beyond supervised learning in federated networks will likely require addressing similar challenges of scalability, heterogeneity, and privacy.

- **Productionizing federated learning.** Beyond the major challenges discussed in this article, there are a number of practical concerns that arise when running federated learning in production. In particular, issues such as concept drift (when the underlying data-generation model changes over time); diurnal variations (when the devices exhibit different behavior at different times of the day or week) [35]; and cold start problems (when new devices enter the network) must be handled with care. We defer the readers to [11], which discusses some of the practical systems-related issues that exist in production federated learning systems.

- **Benchmarks.** Finally, as federated learning is a nascent field, we are at a pivotal time to shape the developments made in this area and ensure that they are grounded in real-world settings, assumptions, and datasets. It is critical for the broader research communities to further build upon existing implementations and benchmarking tools, such as LEAF [16] and TensorFlow Federated [1], to facilitate both the reproducibility of empirical results and the dissemination of new solutions for federated learning.

4 Conclusion

In this article, we have provided an overview of federated learning, a learning paradigm where statistical models are trained at the edge in distributed networks. We have discussed the unique properties and associated challenges of federated learning compared with traditional distributed data center computing and classical privacy-preserving learning. We provided an extensive survey on classical results as well as more recent work specifically focused on federated settings. Finally, we have outlined out a handful of open

problems worth future research effort. Providing solutions to these problems will require interdisciplinary effort from a broad set of research communities.

Acknowledgement. We thank Jeffrey Li and Mikhail Khodak for helpful discussions and comments. This work was supported in part by DARPA FA875017C0141, the National Science Foundation grants IIS1705121 and IIS1838017, an Okawa Grant, a Google Faculty Award, an Amazon Web Services Award, a JP Morgan A.I. Research Faculty Award, a Carnegie Bosch Institute Research Award and the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA, the National Science Foundation, or any other funding agency.

References

[1] Tensorflow federated: Machine learning on decentralized data. URL `https://www.tensorflow.org/federated`.

[2] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Conference on Computer and Communications Security*, 2016.

[3] N. Agarwal, A. T. Suresh, F. X. X. Yu, S. Kumar, and B. McMahan. cpSGD: Communication-efficient and differentially-private distributed sgd. In *Advances in Neural Information Processing Systems*, 2018.

[4] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *International Conference on Management of Data*, 2000.

[5] M. Ammad-ud din, E. Ivannikova, S. A. Khan, W. Oyomno, Q. Fu, K. E. Tan, and A. Flanagan. Federated collaborative filtering for privacy-preserving personalized recommendation system. *arXiv preprint arXiv:1901.09888*, 2019.

[6] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2013.

[7] R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Foundations of Computer Science*, 2014.

[8] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers. Protection against reconstruction and its applications in private federated learning. *arXiv preprint arXiv:1812.00984*, 2018.

[9] T. Bolukbasi, J. Wang, O. Dekel, and V. Saligrama. Adaptive neural networks for efficient inference. In *International Conference on Machine Learning*, 2017.

[10] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. Practical secure aggregation for privacy-preserving machine learning. In *Conference on Computer and Communications Security*, 2017.

[11] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny, S. Mazzocchi, H. B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander. Towards federated learning at scale: system design. In *Conference on Systems and Machine Learning*, 2019.

[12] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. Fog computing and its role in the internet of things. In *SIGCOMM Workshop on Mobile Cloud Computing*, 2012.

[13] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser. Machine learning classification over encrypted data. In *Network and Distributed System Security Symposium*, 2015.

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3:1–122, 2011.

[15] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210*, 2018.

[16] S. Caldas, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[17] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. Song. The secret sharer: Measuring unintended neural network memorization & extracting secrets. *arXiv preprint arXiv:1802.08232*, 2018.

[18] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

[19] M. Castro, B. Liskov, et al. Practical byzantine fault tolerance. In *Operating Systems Design and Implementation*, 1999.

[20] Z. Charles and D. Papailiopoulos. Gradient coding using the stochastic block model. In *International Symposium on Information Theory*, 2018.

[21] Z. B. Charles, D. S. Papailiopoulos, and J. Ellenberg. Approximate gradient coding via sparse random graphs. *arXiv preprint arXiv:1711.0677*, 2017.

[22] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12:1069–1109, 2011.

[23] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.

[24] F. Chen, Z. Dong, Z. Li, and X. He. Federated meta-learning for recommendation. *arXiv preprint arXiv:1802.07876*, 2018.

[25] V. Chen, V. Pastro, and M. Raykova. Secure computation for machine learning with spdz. *arXiv preprint arXiv:1901.00329*, 2019.

[26] L. Corinzia and J. M. Buhmann. Variational federated multi-task learning. *arXiv preprint arXiv:1906.06268*, 2019.

[27] W. Dai, A. Kumar, J. Wei, Q. Ho, G. Gibson, and E. P. Xing. High-performance distributed ML at scale through parameter server consistency models. In *AAAI Conference on Artificial Intelligence*, 2015.

[28] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13:165–202, 2012.

[29] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong. Model-based approximate querying in sensor networks. *The VLDB Journal*, 14:417–443, 2005.

[30] J. Duchi, M. I. Jordan, and B. McMahan. Estimation, optimization, and parallelism when data is sparse. In *Advances in Neural Information Processing Systems*, 2013.

[31] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Privacy aware learning. In *Advances in Neural Information Processing Systems*, 2012.

[32] C. Dwork. A firm foundation for private data analysis. *Communications of the ACM*, 54:86–95, 2011.

[33] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9:211–407, 2014.

[34] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, 2006.

[35] H. Eichner, T. Koren, H. B. McMahan, N. Srebro, and K. Talwar. Semi-cyclic stochastic gradient descent. In *International Conference on Machine Learning*, 2019.

[36] K. El Emam and F. K. Dankar. Protecting privacy using k-anonymity. *Journal of the American Medical Informatics Association*, 15:627–637, 2008.

[37] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *Conference on Knowledge Discovery and Data Mining*, 2004.

[38] V. Feldman, I. Mironov, K. Talwar, and A. Thakurta. Privacy amplification by iteration. In *Foundations of Computer Science*, 2018.

[39] M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Conference on Computer and Communications Security*, 2015.

[40] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere. Edge-centric computing: Vision and challenges. *SIGCOMM Computer Communication Review*, 45:37–42, 2015.

[41] R. C. Geyer, T. Klein, and M. Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[42] B. Ghazi, R. Pagh, and A. Velingker. Scalable and differentially private distributed aggregation in the shuffled model. *arXiv preprint arXiv:1906.08320*, 2019.

[43] S. Goryczka and L. Xiong. A comprehensive comparison of multiparty secure additions with differential privacy. *IEEE Transactions on Dependable and Secure Computing*, 14:463–477, 2015.

[44] N. Guha and V. Smith. Model aggregation via good-enough model spaces. *arXiv preprint arXiv:1805.07782*, 2018.

[45] N. Guha, A. Talwalkar, and V. Smith. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019.

[46] A. Hard, K. Rao, R. Mathews, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[47] L. He, A. Bian, and M. Jaggi. Cola: Decentralized linear learning. In *Advances in Neural Information Processing Systems*, 2018.

[48] Q. Ho, J. Cipar, H. Cui, S. Lee, J. K. Kim, P. B. Gibbons, G. A. Gibson, G. Ganger, and E. P. Xing. More effective distributed ML via a stale synchronous parallel parameter server. In *Advances in Neural Information Processing Systems*, 2013.

[49] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe. Mobile fog: A programming model for large-scale applications on the internet of things. In *SIGCOMM Workshop on Mobile Cloud Computing*, 2013.

[50] J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z. M. Mao, S. Sen, and O. Spatscheck. An in-depth study of lte: effect of network protocol and application behavior on performance. *SIGCOMM Computer Communication Review*, 43: 363–374, 2013.

[51] L. Huang and D. Liu. Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records. *arXiv preprint arXiv:1903.09296*, 2019.

[52] L. Huang, Y. Yin, Z. Fu, S. Zhang, H. Deng, and D. Liu. Loadaboost: Loss-based adaboost federated machine learning on medical data. *arXiv preprint arXiv:1811.12629*, 2018.

[53] R. Iyengar, J. P. Near, D. Song, O. Thakkar, A. Thakurta, and L. Wang. Towards practical differentially private convex optimization. In *Conference on Computer and Communications Security*, 2019.

[54] M. Jaggi, V. Smith, M. Takác, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan. Communication-efficient distributed dual coordinate ascent. In *Advances in Neural Information Processing Systems*, 2014.

[55] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.

[56] P. Jiang and G. Agrawal. A linear speedup analysis of distributed deep learning with sparse and quantized communication. In *Advances in Neural Information Processing Systems*, 2018.

[57] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y.-C. Liang, and D. I. Kim. Incentive design for efficient federated learning in mobile networks: A contract theory approach. *arXiv preprint arXiv:1905.07479*, 2019.

[58] M. Khodak, M.-F. Balcan, and A. Talwalkar. Adaptive gradient-based meta-learning methods. *arXiv preprint arXiv:1906.02717*, 2019.

[59] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[60] T. Kuflik, J. Kay, and B. Kummerfeld. Challenges and solutions of ubiquitous user modeling. In *Ubiquitous Display Environments*. 2012.

[61] A. Lalitha, X. Wang, O. Kilinc, Y. Lu, T. Javidi, and F. Koushanfar. Decentralized bayesian learning over graphs. *arXiv preprint arXiv:1905.10466*, 2019.

[62] C.-P. Lee and D. Roth. Distributed box-constrained quadratic optimization for dual linear svm. In *International Conference on Machine Learning*, 2015.

[63] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran. Speeding up distributed machine learning using codes. *IEEE Transactions on Information Theory*, 64:1514–1529, 2017.

[64] J. Li, M. Khodak, S. Caldas, and A. Talwalkar. Differentially-private gradient-based meta-learning. *Technical Report*, 2019.

[65] T. Li, A. K. Sahu, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith. Federated optimization for heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[66] T. Li, M. Sanjabi, and V. Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

[67] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 2017.

[68] T. Lin, S. U. Stich, and M. Jaggi. Don't use large mini-batches, use local sgd. *arXiv preprint arXiv:1808.07217*, 2018.

[69] Y. Lindell and B. Pinkas. Privacy preserving data mining. In *Advances in Cryptology*, 2000.

[70] L. Liu, J. Zhang, S. Song, and K. B. Letaief. Edge-assisted hierarchical federated learning with non-iid data. *arXiv preprint arXiv:1905.06641*, 2019.

[71] Y. Liu, J. K. Muppala, M. Veeraraghavan, D. Lin, and M. Hamdi. *Data center networks: Topologies, architectures and fault-tolerance characteristics*. Springer Science & Business Media, 2013.

[72] C. Ma, V. Smith, M. Jaggi, M. I. Jordan, P. Richtárik, and M. Takáč. Adding vs. averaging in distributed primal-dual optimization. In *International Conference on Machine Learning*, 2015.

[73] L. W. Mackey, M. I. Jordan, and A. Talwalkar. Divide-and-conquer matrix factorization. In *Advances in Neural Information Processing Systems*, 2011.

[74] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tinydb: an acquisitional query processing system for sensor networks. *Transactions on Database Systems*, 30:122–173, 2005.

[75] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Conference on Artificial Intelligence and Statistics*, 2017.

[76] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.

[77] L. Melis, G. Danezis, and E. D. Cristofaro. Efficient private statistics with succinct sketches. In *Network and Distributed System Security Symposium*, 2016.

[78] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *IEEE Symposium on Security & Privacy*, 2019.

[79] P. Mohassel and P. Rindal. Aby 3: a mixed protocol framework for machine learning. In *Conference on Computer and Communications Security*, 2018.

[80] M. Mohri, G. Sivek, and A. T. Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, 2019.

[81] M. E. Nergiz and C. Clifton. δ-presence without complete world knowledge. *IEEE Transactions on Knowledge and Data Engineering*, 22:868–883, 2010.

[82] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft. Privacy-preserving ridge regression on hundreds of millions of records. In *Symposium on Security and Privacy*, 2013.

[83] T. Nishio and R. Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. In *International Conference on Communications*, 2019.

[84] A. Pantelopoulos and N. G. Bourbakis. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Transactions on Systems, Man, and Cybernetics*, 40:1–12, 2010.

[85] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations*, 2017.

[86] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson. Scalable private learning with pate. In *International Conference on Learning Representations*, 2018.

[87] A. Qiao, B. Aragam, B. Zhang, and E. Xing. Fault tolerance in iterative-convergent machine learning. In *International Conference on Machine Learning*, 2019.

[88] Z. Qu, P. Richtárik, and T. Zhang. Quartz: Randomized dual coordinate ascent with arbitrary sampling. In *Advances in Neural Information Processing Systems*, 2015.

[89] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays. Federated learning for emoji prediction in a mobile keyboard. *arXiv preprint arXiv:1906.04329*, 2019.

[90] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, 2016.

[91] A. Ratner et al. SysML: The new frontier of machine learning systems. *arXiv preprint arXiv:1904.03257*, 2019.

[92] B. Recht, C. Re, S. Wright, and F. Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 2011.

[93] S. J. Reddi, J. Konečnỳ, P. Richtárik, B. Póczós, and A. Smola. Aide: Fast and communication efficient distributed optimization. *arXiv preprint arXiv:1608.06879*, 2016.

[94] A. Reisizadeh, S. Prakash, R. Pedarsani, and A. S. Avestimehr. Coded computation over heterogeneous clusters. *IEEE Transactions on Information Theory*, 65:4227–4242, 2019.

[95] M. S. Riazi, C. Weinert, O. Tkachenko, E. M. Songhori, T. Schneider, and F. Koushanfar. Chameleon: A hybrid secure computation framework for machine learning applications. In *Asia Conference on Computer and Communications Security*, 2018.

[96] P. Richtárik and M. Takáč. Distributed coordinate descent method for learning with big data. *Journal of Machine Learning Research*, 17:2657–2681, 2016.

[97] B. D. Rouhani, M. S. Riazi, and F. Koushanfar. Deepsecure: Scalable provably-secure deep learning. In *Design Automation Conference*, 2018.

[98] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah. Federated learning for ultra-reliable low-latency v2v communications. In *Global Communications Conference*, 2018.

[99] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek. Robust and communication-efficient federated learning from non-iid data. *arXiv preprint arXiv:1903.02891*, 2019.

[100] M. Schmidt and N. L. Roux. Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv preprint arXiv:1308.6370*, 2013.

[101] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *International Speech Communication Association*, 2014.

[102] S. Shalev-Shwartz and T. Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In *Advances in Neural Information Processing Systems*, 2013.

[103] O. Shamir and N. Srebro. Distributed stochastic optimization and learning. In *Allerton Conference on Communication, Control, and Computing*, 2014.

[104] O. Shamir, N. Srebro, and T. Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *International Conference on Machine Learning*, 2014.

[105] S. Silva, B. Gutman, E. Romero, P. M. Thompson, A. Altmann, and M. Lorenzi. Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data. *arXiv preprint arXiv:1810.08553*, 2018.

[106] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, 2017.

[107] V. Smith, S. Forte, C. Ma, M. Takac, M. I. Jordan, and M. Jaggi. Cocoa: a general framework for communication-efficient distributed optimization. *Journal of Machine Learning Research*, 18:1–47, 2018.

[108] S. U. Stich. Local sgd converges fast and communicates little. In *International Conference on Learning Representations*, 2019.

[109] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis. Gradient coding: Avoiding stragglers in distributed learning. In *International Conference on Machine Learning*, 2017.

[110] A. S. Tanenbaum and M. Van Steen. *Distributed systems: principles and paradigms*. Prentice-Hall, 2007.

[111] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu. Communication compression for decentralized training. In *Advances in Neural Information Processing Systems*, 2018.

[112] H. Tang, C. Yu, C. Renggli, S. Kassing, A. Singla, D. Alistarh, J. Liu, and C. Zhang. Distributed learning over unreliable networks. In *International Conference on Machine Learning*, 2019.

[113] O. Thakkar, G. Andrew, and H. B. McMahan. Differentially private learning with adaptive clipping. *arXiv preprint arXiv:1905.03871*, 2019.

[114] S. Thrun and L. Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

[115] C. Van Berkel. Multi-core for mobile phones. In *Conference on Design, Automation and Test in Europe*, 2009.

[116] S. Vaswani, F. Bach, and M. Schmidt. Fast and faster convergence of sgd for over-parameterized models (and an accelerated perceptron). In *Conference on Artificial Intelligence and Statistics*, 2019.

[117] P. Vepakomma, O. Gupta, A. Dubey, and R. Raskar. Reducing leakage in distributed deep learning for sensitive health data. *arXiv preprint arXiv:1812.00564*, 2019.

[118] I. Wagner and D. Eckhoff. Technical privacy metrics: a systematic survey. *ACM Computing Surveys*, 51:57, 2018.

[119] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright. Atomo: Communication-efficient learning via atomic sparsification. In *Advances in Neural Information Processing Systems*, 2018.

[120] J. Wang and G. Joshi. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. *arXiv preprint arXiv:1808.07576*, 2018.

[121] J. Wang and G. Joshi. Adaptive communication strategies to achieve the best error-runtime trade-off in local-update sgd. In *Conference on Systems and Machine Learning*, 2019.

[122] S. Wang, F. Roosta-Khorasani, P. Xu, and M. W. Mahoney. Giant: Globally improved approximate newton method for distributed optimization. In *Advances in Neural Information Processing Systems*, 2018.

[123] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan. Adaptive federated learning in resource constrained edge computing systems. *Journal on Selected Areas in Communications*, 37:1205–1221, 2019.

[124] WeBank AI Group. Federated learning white paper v1.0. 2018.

[125] B. Woodworth, J. Wang, A. Smith, B. McMahan, and N. Srebro. Graph oracle models, lower bounds, and gaps for parallel stochastic optimization. In *Advances in Neural Information Processing Systems*, 2018.

[126] X. Wu, F. Li, A. Kumar, K. Chaudhuri, S. Jha, and J. Naughton. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *International Conference on Management of Data*, 2017.

[127] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10:12, 2019.

[128] T. Yang. Trading computation for communication: Distributed stochastic dual coordinate ascent. In *Advances in Neural Information Processing Systems*, 2013.

[129] Y. Yao, L. Rosasco, and A. Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26:289–315, 2007.

[130] D. Yin, A. Pananjady, M. Lam, D. Papailiopoulos, K. Ramchandran, and P. Bartlett. Gradient diversity: a key ingredient for scalable distributed learning. In *Conference on Artificial Intelligence and Statistics*, pages 1998–2007, 2018.

[131] H. Yu, S. Yang, and S. Zhu. Parallel restarted sgd for non-convex optimization with faster convergence and less communication. In *AAAI Conference on Artificial Intelligence*, 2018.

[132] H. Yu, R. Jin, and S. Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *International Conference on Machine Learning*, 2019.

[133] J. Yuan and S. Yu. Privacy preserving back-propagation neural network learning made practical with cloud computing. *IEEE Transactions on Parallel and Distributed Systems*, 25:212–221, 2013.

[134] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, T. N. Hoang, and Y. Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, 2019.

[135] H. Zhang, J. Li, K. Kara, D. Alistarh, J. Liu, and C. Zhang. ZipML: Training linear models with end-to-end low precision, and a little bit of deep learning. In *International Conference on Machine Learning*, 2017.

[136] S. Zhang, A. E. Choromanska, and Y. LeCun. Deep learning with elastic averaging sgd. In *Advances in Neural Information Processing Systems*, 2015.

[137] Y. Zhang, J. Duchi, and M. Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research*, 16:3299–3340, 2015.

[138] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[139] Y. Zhao, J. Zhao, L. Jiang, R. Tan, and D. Niyato. Mobile edge computing, blockchain and reputation-based crowdsourcing iot federated learning: A secure, decentralized and privacy-preserving system. *arXiv preprint arXiv:1906.10893*, 2019.

[140] F. Zhou and G. Cong. On the convergence properties of a *k*-step averaging stochastic gradient descent algorithm for nonconvex optimization. In *International Joint Conference on Artificial Intelligence*, 2018.

[141] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 2010.